

# Toetsing van de pilot koppelvlakken RSGB bevestigingen nieuwe stijl

Tussentijdse beoordeling van de opgeleverde standaard (dd. 30 juni 2016)

Liam Blythe, Niels van der Zwan

5 Juli 2016



GETTING SOFTWARE RIGHT

# Inhoudsopgave

- 1 Randvoorwaarden
  - 2 Toetsingscriteria
  - 3 Beoordeling
  - 4 Bevindingen
-

# Randvoorwaarden

## Uitgangspunten pilot groep

- De functionele behoeften van gebruikers is stabiel
- De nadruk komt te liggen op de ontwikkeling van eenduidige interfaces
- Het volgen van standaarden is gewenst, maar ondergeschikt aan de eenduidigheid

# Randvoorwaarden

## Uitgangspunten van SIG

- Nadruk ligt op de bruikbaarheid van de standaard vanuit het perspectief van de gebruikers; de partijen die o.b.v. de standaard software schrijven om uitwisseling van data te faciliteren.
  - Hefboom: Eenvoudig gebruik voor een grote groep gebruikers afwegen tegen (eventueel) extra onderhoudslast voor een kleine groep beheerders
  - SIG beschouwt het beheerdersperspectief van het koppelvlak als ondergeschikt aan de behoeften van de (gemeentelijke) gebruikers.
  
- Uitgangspunten SIG
  - Een kwalitatief goede standaard moet eenvoudig zijn in het gebruik
  - Eenvoudigheid in gebruik faciliteert het schrijven van kwalitatief goede software
  - Kwalitatief goede software vermindert de onderhoudslast en kosten.

# Inhoudsopgave

- 1 Randvoorwaarden
  - 2 Toetsingscriteria
  - 3 Beoordeling
  - 4 Bevindingen
-

# Toetsingscriteria voor een standaard

Uit het SIG Rapport (2015)

## SIG hanteert de volgende toetsingscriteria m.b.t. standaarden

- 1. Geaccepteerd:** Door een autoriteit en/of een groep belanghebbenden
- 2. Gangbaar:** Het gebruik wordt algemeen als normaal beschouwd
- 3. Toepasselijk:** Toepasbaar en compleet binnen een afgebakend domein of probleem set
- 4. Bruikbaar:** Wordt ervaren als eenvoudig in gebruik. Vermindert de inspanning/drempel om de standaard te gebruiken.
- 5. Volwassen:** Voldoende ondersteunende tooling, ervaring, documentatie. Referentie/Proef implementatie(s)
- 6. Stabiel:** Gestructureerd verbeteringsproces, wijzigingen zijn backward compatibel, impact van wijzigingen is minimaal

Huidige fase van ontwikkeling van RSGB bevestigingen koppelvak maakt dit criterium prematuur

Het RSGB bevestigingen koppelvak is nog niet in (operationeel) gebruik

Buiten scope van opdracht SIG

# Criteria voor een goede standaard

Op basis van toetsingscriteria uit het SIG rapport

## Bruikbaar (4)

- > Beperkte **omvang**
- > Beperkte **complexiteit**
- > **Eenduidige** schema's
- > **Consistente** schema's

## Volwassen (5)

- > Ondersteund door **tooling**
- > Beschikbaarheid van **referentie/proef implementatie(s)**
- > Toetsbare **compliance** van implementaties

## Stabiel (6)

- > Stabiliteit van **functionele** specificaties
- > Stabiliteit van **technische** implementatie

# Toetsingscriteria

## Criteria voor het beoordelen van de bruikbaarheid (1/2)

### Omvang

- › Meting van de totale volume van het koppelvlak
- › Beïnvloedt de benodigde inspanning om een koppelvlak te
  - Begrijpen
  - Implementeren
- › Telling van het aantal regels code (lines of code – LOC)

### Complexiteit

- › Meting van de structurele variëteit die kan optreden in het koppelvlak
- › Beïnvloedt de benodigde inspanning om een koppelvlak te implementeren
- › Telling van de McCabe complexiteit, relatief aan de omvang



# Toetsingscriteria

## Criteria voor het beoordelen van de bruikbaarheid (2/2)

### Eenduidigheid

- › De mate waarin het koppelvlak het correct gebruik beschrijft en afdwingt zonder noodzaak voor aanvullende documentatie.
- › Beïnvloedt de benodigde inspanning om een koppelvlak te begrijpen
- › Voorkomt uiteenlopende functionele implementaties.

### Consistentie

- › De mate waarin vergelijkbare concepten en functionaliteiten volgens dezelfde conventies worden vastgelegd
- › Dit vergemakkelijkt voor afnemer en provider:
  - Het begrijpen van het koppelvlak
  - De implementatie van een koppelvlak

# Toetsingscriteria

## Criteria voor het beoordelen van de volwassenheid (1/2)

### Ondersteund door Tooling

- > **Object Oriëntatie:** Koppelvlak schema's zijn object georiënteerd, zodat de meest gebruikte moderne (object georiënteerde) ontwikkeltalen de schema's optimaal kunnen implementeren.
- > **Code generatie:** Code generatoren (Java en .NET) kunnen de gebruikte constructies in de koppelvlak schema's volledig vastleggen.

# Toetsingscriteria

## Criteria voor het beoordelen van de volwassenheid (2/2)

### Referentie/Proef Implementatie(s)

Een referentie/proef implementatie is een (versimpelde) applicatie die het functioneel juist gebruik van de standaard demonstreert

#### > **Aanwezigheid:**

- Voor meerdere (meest gebruikte) ontwikkeltalen (minimaal Java en .NET)
- Voor alle versies van een koppelvlak

#### > **Beschikbaarheid:** Aanwezige referentie implementaties zijn openlijk en gratis beschikbaar.

# Toetsingscriteria

## Criteria voor het beoordelen van de stabiliteit

### Stabiliteit van functionele specificaties

- De mate waarin de functionele specificaties gevoelig zijn voor wijzigingen
- Functionele specificaties die gevoelig zijn voor wijziging verhogen de kans dat een koppelvlak en/of interface moet worden aangepast

### Stabiliteit van technische implementatie

- De impact die een wijziging heeft voor implementaties van het koppelvlak

# Inhoudsopgave

- 1 Randvoorwaarden
  - 2 Toetsingscriteria
  - 3 Beoordeling
  - 4 Bevindingen
-

# Beoordeling

Aanpak voor het beoordelen van de verschillende criteria

Toetsingscriterium	Eigenschap	Toetsingsmethode
Bruikbaarheid	Omvang	SIG Benchmark
	Complexiteit	SIG Benchmark
	Eenduidigheid	Review door SIG
	Consistentie	Review door SIG
Volwassenheid	Ondersteund door tooling	Review door SIG
	Referentie Implementatie(s)	Review door SIG
Stabiliteit	Functioneel	Externe input
	Technisch	Review door SIG

# Beoordeling

## Toetsingscriteria

Toetsingscriterium	Aspect	Oordeel	Bevindingen
Bruikbaarheid	Beperkte Omvang	Ok	• Beperkte omvang: Kleinste standaard in de SIG benchmark (omvang is 635 lines of code)
	Beperkte Complexiteit	Ok	• Lage complexiteit: op 3 na de minst complexe standaard in de SIG benchmark
	Eenduidig	Aandacht nodig	• Het koppelvlak heeft aandachtspunten, maar geen structurele problemen <ul style="list-style-type: none"> <li>• Berichtstructuren en parameter namen zijn eenduidig</li> <li>• Standaard en geldige waarden en berichtnamen zijn niet volledig eenduidig</li> </ul>
	Consistent	Verbetering noodzakelijk	• Naamgevingen zijn wel consistent <ul style="list-style-type: none"> <li>• Vormgeving van relaties is niet consistent</li> <li>• Eigenschappen van verblijfplaats en verblijfadres hetzelfde, maar andere benamingen</li> <li>• Term “persoon” dekt meer dan “natuurlijke persoon”</li> </ul>
Volwassenheid	Ondersteunde tooling	Ok	• Schema's volgen paradigma van object oriëntatie (inheritance relaties ipv choices, geen complexType restrictions) <ul style="list-style-type: none"> <li>• Gebruikte constructies worden grotendeels ondersteund door code generatoren</li> </ul>
	Referentie implementaties	Aandacht nodig	• Java en C#, providers en consumers zijn in ontwikkeling <ul style="list-style-type: none"> <li>• Referentie implementaties (nog) niet openlijk beschikbaar</li> </ul>
Stabiliteit	Functioneel	Ok	• Pilot groep heeft aangegeven dat functionele behoefte in de afgelopen 10 jaar vrijwel niet is gewijzigd
	Technisch	Verbetering noodzakelijk	• De specifieke operaties verhogen de gevoeligheid voor breaking change bij functionele wijzigingen <ul style="list-style-type: none"> <li>• Versionerings-strategie is nog niet gedefinieerd</li> </ul>



Ok



Aandacht nodig



Verbetering noodzakelijk

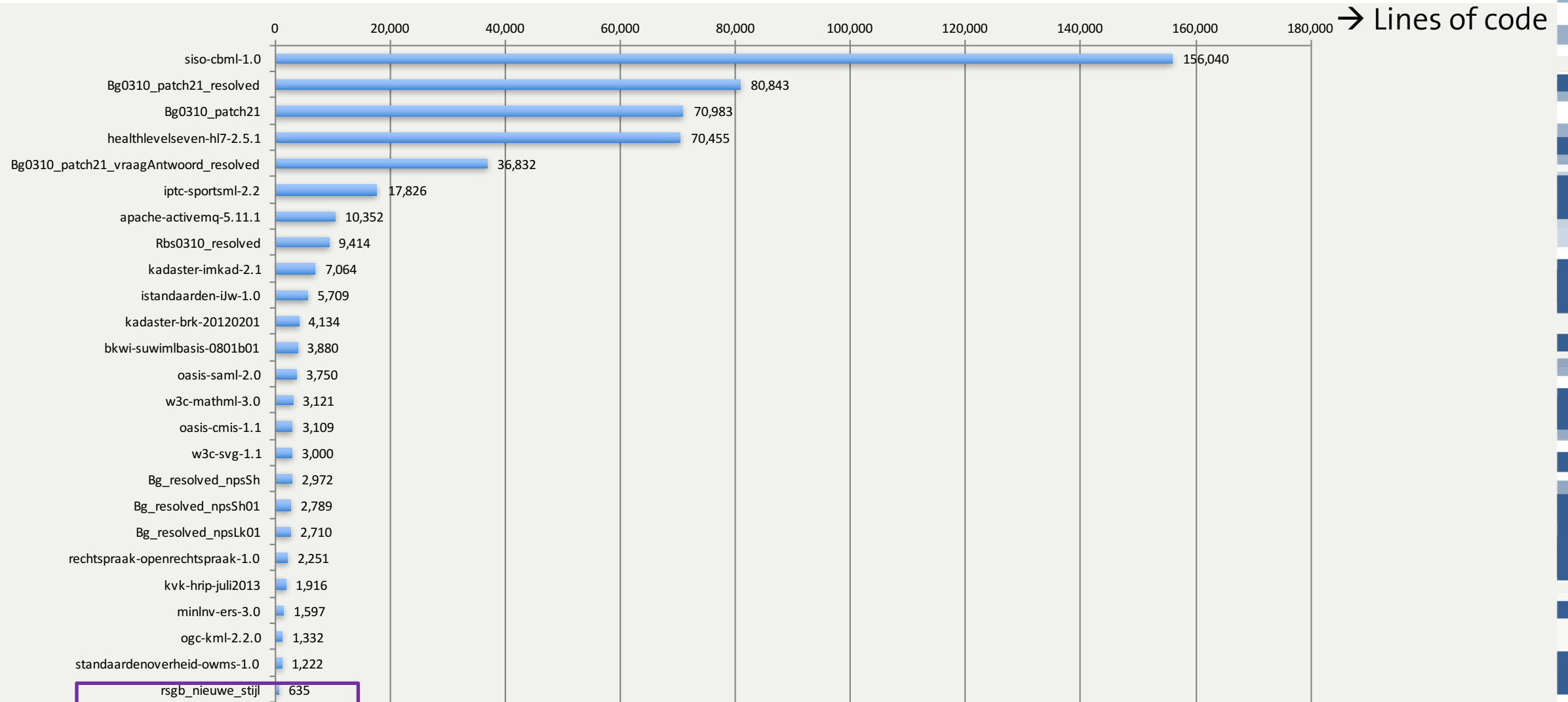
# Inhoudsopgave

- 1 Randvoorwaarden
  - 2 Toetsingscriteria
  - 3 Beoordeling
  - 4 Bevindingen
-



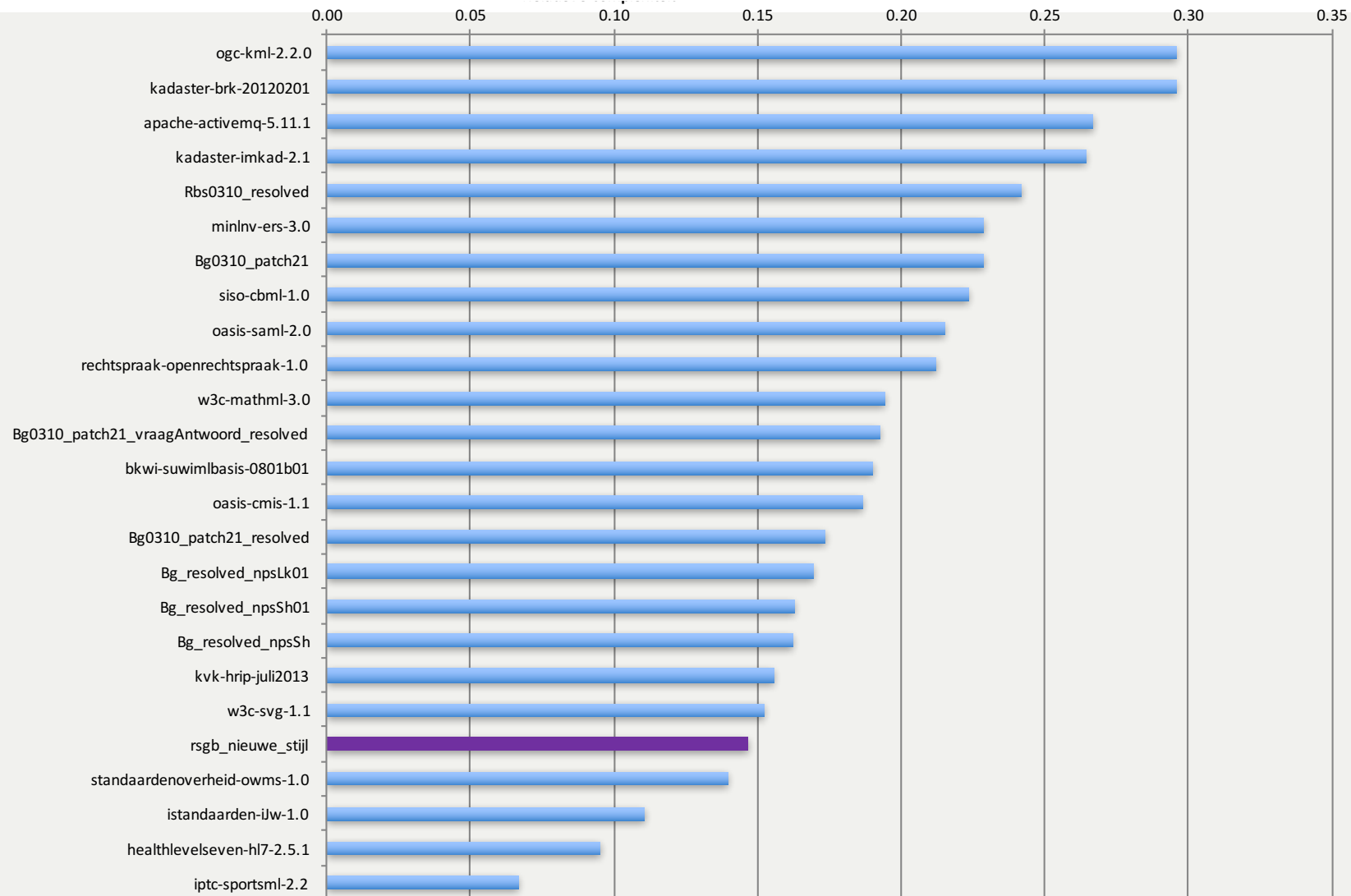
# Omvang

Het koppelvlak is de kleinste standaard in de SIG benchmark



# Complexiteit

Het koppelvlak is één van de minst complexe standaarden in de SIG benchmark



→ Relatieve complexiteit

# Eenduidigheid

## Criteria voor het beoordelen van eenduidigheid

### SIG gebruikt de volgende criteria ter beoordeling van de eenduidigheid

- > **Structurele eenduidigheid:** De schema's dwingen geldige berichten af
  - **Hoe:** Opgenomen elementen en waarden die volgens het schema valide zijn, zijn ook geldig zijn in het gebruik
  
- > **Functionele eenduidigheid:** Het beoogd functioneel gebruik wordt door de schema's omschreven.
  - **Hoe:** Het verband tussen vraag en antwoord is afleidbaar, doordat
    - Namen van vraagberichten expliciet het beoogd gedrag beschrijven  
Voorbeeld. *ZoekIngeschrevenPersoonOpGeslachtsnaam* om ingeschreven personen o.b.v. Het geslachtsnaam te bevragen
    - Parameter namen overeenkomen met eigenschappen van bevraagde entiteiten  
Voorbeeld. *geslachtsnaam*, *postcode*, etc.
    - Parameter namen gerelateerd kunnen worden aan eigenschappen van bevraagde entiteiten  
Voorbeeld. De parameter *inclusiefOverledenPersonen* is eenduidig te relateren aan de eigenschap “*overlijdensDatum*”

# Eenduidigheid

Structurele eenduidigheid: Berichtstructuur is eenduidig

- Het is mogelijk om bij “persoon” een volledige **NatuurlijkePersoon** op te nemen.
- Het koppelvlak belooft dat er een **NatuurlijkePersoonBeperkt** opgenomen zal worden
- Dit is functioneel geldig, ongeacht of het basistype of specialisatie wordt opgenomen
- Een afnemer die een basistype verwacht, maar een specialisatie ontvangt, ondervindt hier geen problemen aan

```
<complexType name="Kind">
  <complexContent>
    <extension base="STUF:Relatie">
      <sequence>
        <element name="persoon" type="STUF:NatuurlijkPersoonBeperkt" />
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

----->

```
<complexType name="NatuurlijkPersoonBeperkt">
  <sequence>
```

↑

```
<complexType name="NatuurlijkPersoon">
  <complexContent>
    <extension base="STUF:NatuurlijkPersoonBeperkt">
```

# Eenduidigheid

Structurele eenduidigheid: Restricties op geldige waarden nog onvolledig

- SimpleType restrictions worden gebruikt om de "range" van geldige waarden te specificeren
- Niet alle simpleTypes specificeren volledig de "range" van geldige waarden. Voorbeeld:
  - **EmailAdres** dwingt niet af dat er een "@" voorkomt

```
<simpleType name="Geslachtsaanduiding">
  <restriction base="string">
    <enumeration value="Man"/>
    <enumeration value="Vrouw"/>
    <enumeration value="Onbekend"/>
  </restriction>
</simpleType>

<simpleType name="Huisletter">
  <restriction base="string">
    <pattern value="[a-zA-Z]{1}" />
  </restriction>
</simpleType>
```

```
<simpleType name="Emailadres">
  <restriction base="string">
    <maxLength value="254"/>
  </restriction>
</simpleType>
```

# Eenduidigheid

Functionele eenduidigheid: Naamgeving bij berichten is niet volledig eenduidig

- Verblijfsadres specificeert een abstract type **Adres**.
- De bevraging **ZoekIngeschrevenPersoonOpVerblijfsadres** specificeert parameters voor het zoeken op binnenlandse adressen (een specialisatie van **Adres**). Dit is **niet** eenduidig
- Dit is een kwestie van benaming van de bevraging: Bijv. **ZoekIngeschrevenPersoonOpBinnenlandsVerblijfsadres** bakent deze bevraging af

```
<<element name="ZoekIngeschrevenPersoonOpVerblijfsadres">
  <<annotation>
    <documentation>Met de operatie is het niet mogelijk om naar niet-ingezet
  </annotation>
  <complexType>
    <sequence>
      <element name="woonplaatsnaam" type="STUF:PlaatsNaam" minOccurs="1" />
      <element name="openbareRuimteNaam" minOccurs="1">
        <simpleType>
          <restriction base="string">
            <pattern value="[%]{0,1}[a-zA-Z'-]{2,200}[%]{0,1}" />
          </restriction>
        </simpleType>
      </element>
      <element name="huisnummer" type="STUF:Huisnummer" minOccurs="1" />
      <element name="huisletter" type="STUF:Huisletter" />
      <element name="huisnummertoevoeging" type="STUF:Huisnummertoevoeging" />
      <element name="locatieaanduiding" type="STUF:Locatieaanduiding" />
      <element name="periode" type="STUF:Ingangsdatum">

```

```
<complexType name="Verblijfsadres">
  <sequence>
    <element name="adres" type="STUF:Adres" minOccurs="1" />
    <element name="inOnderzoek" type="STUF:InOnderzoek" minOccurs="0" />
    <element name="ingangsdatum" type="STUF:Ingangsdatum" minOccurs="1" />
  </sequence>
</complexType>

```

# Consistentie

## Criteria voor het beoordelen van consistentie

### SIG gebruikt de volgende criteria ter beoordeling van de consistentie

- **Structurele consistentie:** Structuren hanteren afleidbare conventies.
  - **Hoe:**
    - Conventies voor structuren zijn afleidbaar bij Operaties, Berichten, Parameters, Typen  
(entiteit/relatie/bericht/waarden)
    - Conventies voor naamgevingen zijn afleidbaar bij Operaties, Berichten, Parameters en Typen  
(entiteit/relatie/bericht/waarden)
- **Functionele consistentie:** Beoogde functionaliteiten zijn op soortgelijke wijzen vormgegeven
  - **Hoe:**
    - Conventies voor vormgeving van beoogde functionaliteit is afleidbaar
    - Voor dezelfde typen worden dezelfde termen gebruikt

# Consistentie

Structureel: Schema's maken geen duidelijk onderscheid tussen Entiteiten en Relaties

➤ SIG herkent in de entiteiten schema's van het koppelvlak drie basisconcepten:

- Entiteiten
- Relaties
- Waarden

➤ Uit de naamgevingsconventies is niet af te leiden of **Medebewoner**, **Verblijfplaats** en **Verblijfsadres** entiteiten of relaties zijn:

- De benaming van deze types geeft hiertoe geen aanwijzing
- Deze types leiden niet af van “**Relatie**”, dan wel een “**Beperkt**” entiteit

➤ SIG doet de aanname dat bovenstaande relaties zijn

```
<complexType name="Relatie" abstract="true">
  <sequence>
    <element name="periode" type="STUF:Ingangsdatum" minOccurs="1" />
  </sequence>
</complexType>
```

*Ouder leidt af van Relatie*

```
<complexType name="Ouder">
  <complexContent>
    <extension base="STUF:Relatie">
      <sequence>
        <element name="persoon" type="STUF:NatuurlijkPersoonBeperkt" />
      </sequence>
    </extension>
  </complexContent>
```

*Verblijfplaats leidt niet af van Relatie*

```
<complexType name="Verblijfplaats">
  <sequence>
    <element name="woonadres" type="STUF:Adres" minOccurs="1" />
    <element name="periode" type="STUF:Ingangsdatum" minOccurs="1" />
  </sequence>
</complexType>
```

```
<complexType name="Verblijfsadres">
  <sequence>
    <element name="adres" type="STUF:Adres" minOccurs="1" />
    <element name="InOnderzoek" type="STUF:InOnderzoek" minOccurs="0" />
    <element name="Ingangsdatum" type="STUF:Ingangsdatum" minOccurs="1" />
  </sequence>
</complexType>
```

```
<complexType name="Medebewoner">
  <sequence>
    <element name="Ingangsdatum" type="date" minOccurs="1" />
    <element name="persoon" type="STUF:NatuurlijkPersoonBeperkt" minOccurs="1" />
  </sequence>
</complexType>
```



# Consistentie

Functioneel: Eigenschappen van **Verblijfplaats** en **Verblijfsadres** verschillende benamingen

- > Eigenschappen **adres** en **woonadres** refereren allebei naar het type **Adres**
- > Eigenschappen **Periode** en **IngangsDatum** refereren allebei naar het type **IngangsDatum**.

```
<complexType name="Verblijfplaats"><sequence><element name="woonadres" type="STUF:Adres" minOccurs="1" /><element name="periode" type="STUF:Ingangsdatum" minOccurs="1" /></sequence></complexType>
```

```
<complexType name="Verblijfsadres"><sequence><element name="adres" type="STUF:Adres" minOccurs="1" /><element name="inonderzoek" type="STUF:inonderzoek" minOccurs="0" /><element name="ingangsdatum" type="STUF:Ingangsdatum" minOccurs="1" /></sequence></complexType>
```

# Ondersteunde tooling

## Object Oriëntatie: Faciliteren hergebruik door inheritance

- Inheritance relaties faciliteren hergebruik van gedeelde eigenschappen
- Hergebruik vermindert duplicatie in schema's en implementaties doordat gedeelde definities en implementatie logica eenmalig worden vastgelegd.
- **Voorbeeld:**
  - Kind, ouder en partner zijn specialisaties van een relatie.
  - Relaties hebben een periode van geldigheid.
  - Eigenschappen voor periode geldigheid zijn opgenomen in het basistype "relatie"

```
<complexType name="Relatie" abstract="true">
  <sequence>
    <element name="periode" type="STUF:Ingangsdatum" minOccurs="1" />
  </sequence>
</complexType>

<complexType name="Ouder">
  <complexContent>
    <extension base="STUF:Relatie">
      <sequence>
        <element name="persoon" type="STUF:NatuurlijkPersoonBeperkt" />
      </sequence>
    </extension>
  </complexContent>
</complexType>

<complexType name="Partner">
  <complexContent>
    <extension base="STUF:Relatie">
      <sequence>
        <element name="persoon" type="STUF:NatuurlijkPersoonBeperkt" />
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

# Ondersteunde tooling

Code Generatie: Volledig voor zover ondersteund door standaard code generatoren

- De volgende – voor code generatoren problematische - constructies komen niet voor in de XML schema's
  - ComplexType restrictions
  - Choice elementen
  - Fixed attributen
  - Combinatie van nillable="true" en minOccurs="0"
- SimpleType restrictions komen WEL voor in de XML schema's. Deze worden niet (altijd) ondersteund door .Net en Java code generatoren, echter:
  - Hebben documenterende waarde, zonder noodzaak van aanvullende documentatie
  - Kunnen wel afgedwongen worden met schema validatie

# Stabiliteit

## Toetsingscriteria voor technische stabiliteit

### SIG hanteert de volgende beoordelingscriteria voor technische stabiliteit:

- > **Impact** van een wijziging aan
  - Het koppelvlak bij wijziging van de functionele specificaties
  - Een implementatie bij wijziging van het koppelvlak
  
- > **Mitigatie** van de impact
  - Op wijzigingen aan het koppelvlak
  - Op wijzigingen aan een implementatie

# Stabiliteit

Impact: Wijzigingen aan de functionele specificaties leiden tot wijziging koppelvak

- > **Afweging:** Specifiek koppelvak vs Generiek koppelvak
- > **Gekozen aanpak:** Zo specifiek mogelijk de functionele specificaties in het koppelvak opnemen
- > **Gevolg van gekozen aanpak:** Wijzigingen aan de functionele specificaties hebben snel wijziging van het koppelvak tot gevolg

# Stabiliteit

Mitigatie: Versionerings-strategie is nog niet gedefinieerd

- Een **versionerings-strategie** maakt de geïmplementeerde versie van het koppelvlak expliciet.  
Dit stelt providers in staat
  - Expliciet aan te geven welke versie(s) van het koppelvlak wordt/worden aangeboden
  - Oude versies op termijn uit te faseren, waarbij afnemers binnen dit termijn hun implementaties (indien noodzakelijk) aan kunnen passen
- Er is echter nog geen versionerings-strategie gedefinieerd

## Contact




+31 20 314 0950



[l.blythe@sig.eu](mailto:l.blythe@sig.eu)



[@sig\\_eu](https://twitter.com/sig_eu)



GETTING SOFTWARE RIGHT